PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-007129
(43)Date of publication of application : 11.01.2002

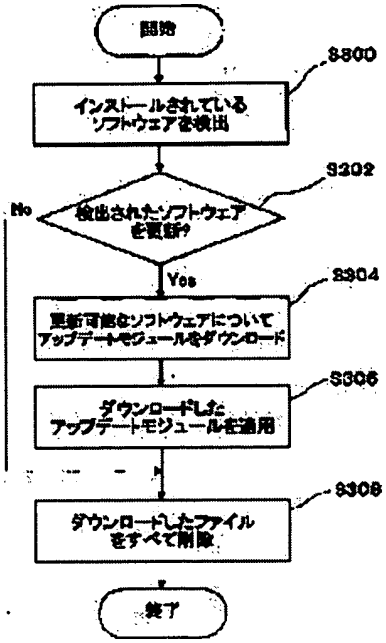
(51)Int.Cl. G06F 9/445
G06F 11/00

(21)Application number : 2000-184282
(22)Date of filing : 20.06.2000

(71)Applicant : OKI ELECTRIC IND CO LTD
(72)Inventor : HORIKAWA SHINICHI

(54) METHOD FOR UPDATING SOFTWARE AND SYSTEM FOR THE SAME

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method and system for updating software capable of updating-software without consuming any storage capacity, and updating software by performing access only to one Web site.
SOLUTION: This method comprises a step for allowing a client system 100 to perform access to a server system 106, and to down-load a check module necessary for the updating processing of software 104, a step (S302) for judging an update module necessary for the updating processing of the software installed in the client system by the check module, a step (S304) for allowing the client system to perform access to the server system and to down-load the update module necessary for the updating processing of each software, and a step (S308) for erasing the check module and the update module from the client system.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] By downloading an update module from a server system to a client system through a network, and applying said update module to said client system The check module for being the approach of updating software installed in said client system, and said client system accessing said server system, and judging an update module required for an update process of software The 1st step to download and the 2nd step which judges an update module required for an update process of the software installed in said client system with said check module, The 3rd step where said client system accesses said server system, and downloads an update module required for an update process of each software, The updating approach of software characterized by including the 4th step which deletes said check module and said update module from the inside of said client system.
[Claim 2] Said 2nd step is the updating approach of software according to claim 1 which said check module investigates the configuration file of software, and is characterized by including the phase of judging whether this software being updated, and the phase which notifies the user of said client system of that when said check module judges that software cannot be updated.
[Claim 3] The updating approach of software according to claim 1 or 2 characterized by including the phase of judging the existence of the computer virus of each software with said check module after said 1st step.
[Claim 4] By downloading an update module from a server system to a client system through a network, and applying said update module to said client system It is the system which updates software installed in said client system. Said server system The 1st storage means which collects and accumulates update information required for renewal of said software from the software vendor system which offers said update module, The 2nd storage means which collects and accumulates an update module applicable to said client system from said software vendor system, A check module offer means to provide said client system with the check module which distinguishes an update module applicable to said client system according to the demand from said client system, ****** -- the updating system of software characterized by things.
[Claim 5] Said check module is an updating system of software according to claim 4 which investigates the configuration file of software and is characterized by having the function to judge whether this software can be updated, and the function which notifies the user of said client system of that when it is judged that an update process of software cannot be performed.
[Claim 6] Said check module is an updating system of software according to claim 4 or 5 characterized by having the function to judge the existence of the computer virus of software.
[Claim 7] Said check module is the updating system of software given in either of claims 4, 5, or 6 which unifies the update information collected from said software vendor system, creates comprehensive update information, and is characterized by having the function which distinguishes an update module applicable to said client system based on this comprehensive update information.
[Claim 8] Said check module is the updating system of software given in either of claims 4, 5, 6, or 7 which is characterized by having the function which downloads an update module applicable to said client system from said server system, and the function which applies this update module

to said client system.

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the updating approach and updating system which update software installed in the computer system (update).
[0002]
[Description of the Prior Art] In recent years, the update module which performs correction, functional enhancement, etc. of software came to be frequently offered with an advancement and complication of a computer system. Although the website on the Internet is widely used as a distribution means of an update module, generally they existed for every software vendor, and have carried the information about an update module for every software product in many cases.
[0003] It is not few also when the software of a software vendor which two or more software is installed in the computer system (a "client system" is called hereafter.) which receives offer of software and its update module on the other hand, and is different is in a dependency. Therefore, in a client system, a great effort is required for choice of the update module which should be applied.
[0004] As a conventional technique which solves such a problem, there are some which are indicated by reference "equipment" which distributes renewal of software through a Seamus DONOHYU:" computer network, and JP,11-272454,A." By the approach shown by this reference, the updater component is installed in the client system with the software used as the object for updating, and renewal of software is performed as follows by this updater component.
[0005] (1) Search a software vendor system to each software installed in the client system through the search engine prepared on the network.
(2) Download an updating list from each website and distinguish the propriety of renewal of each software based on these updating lists.
(3) About the software which can be updated, download an update module from the software vendor system, and apply to a client system.
[0006] By the above-mentioned reference, although the case of the UNIX (trademark) system OS (Operating System) is taken up as an example, "Windows Update" of Microsoft Corp. and "Live Update" of Symantec exist in the technique of the same kind in the Windows (trademark) system OS. Although these [both] are aimed only at the software product which each company offers, renewal of software is performed except for the point of not using a search engine by the same processing as the above-mentioned reference.
[0007]
[Problem(s) to be Solved by the Invention] By the way, when updating two or more software which a different software vendor offers, in the above-mentioned conventional technique, it is necessary to judge the software vendor system which corresponds for every software, and to access it. This is not desirable, considering the viewpoint of effective use of the limited network resource. Moreover, when much software is installed in the client system, the need for renewal of the software of these large number must always be grasped, and there is no disadvantage **** ball.
[0008] With the conventional technique, the point that each updater component in connection

with renewal of software is installed in a client system in advance is common again. This updater component may have to update itself with the advance of the whole client system, and may become existence of only consuming the capacity of storage vainly.
[0009] This invention is made in view of the above-mentioned trouble which the updating system of the conventional software has, and the object of this invention is a thing with the possible renewal of software for which new and the improved updating system of software are offered, without wasting the storage capacity of a client system.
[0010] Moreover, other objects of this invention are things with possible update of the software in a client system for which new and the improved updating system of software are offered by accessing only to one website.
[0011]
[Means for Solving the Problem] In order to solve the above-mentioned technical problem, according to the 1st viewpoint of this invention, like a publication to claim 1 By downloading an update module from a server system to a client system through a network, and applying an update module to a client system The check module for being the approach of updating software installed in the client system, and a client system accessing a server system, and judging an update module required for an update process of software The 1st step to download and the 2nd step which judges an update module required for an update process of the software installed in the client system with a check module, The 3rd step where a client system accesses a server system and downloads an update module required for an update process of each software, The updating approach of the software characterized by including the 4th step which deletes a check module and an update module from the inside of a client system is offered.
[0012] What is necessary is to download a check module from a server system, only when it is not necessary to install the program equivalent to an updater component in a client system in advance and the need for renewal of software arises according to this updating approach. And after an update process of software is completed, all the files relevant to a check module and an update module can be deleted. For this reason, storage capacity of a client system is not consumed vainly.
[0013] Moreover, since all software installed in the client system by accessing one server system can be updated, while mitigating consumption of a network resource, the effort of renewal of software is mitigable.
[0014] Moreover, as for the 2nd step, it is desirable to include the phase of according to claim 2 judging whether a check module being able to investigate the configuration file of software and being able to update this software like, and the phase which notifies the user of a client system of that when a check module judges that software cannot be updated.
[0015] According to this updating approach, a check module investigates the configuration file of software, and even if there is nonconformity -- the indispensable file is lost by software -- it is restorable with the worst user's own hand. For this reason, the software in a client system can be maintained at an always healthy condition.
[0016] Moreover, it is desirable to include the phase according to claim 3 of judging the existence of the computer virus of each software, with a check module after the 1st step like.
[0017] According to this updating approach, it is automatically removable even if computer virus exists in the software of a client system. For this reason, the software in a client system can be maintained at an always healthy condition.
[0018] In order to solve the above-mentioned technical problem, according to the 2nd viewpoint of this invention, moreover, like a publication to claim 4 By downloading an update module from a server system to a client system through a network, and applying an update module to a client system It is the system which updates software installed in the client system. A server system The 1st storage means which collects and accumulates update information required for renewal of software from the software vendor system which offers an update module, The 2nd storage means which collects and accumulates an update module applicable to a client system from a software vendor system, An update module applicable to a client system The updating system of the software characterized by including a check module offer means to provide a client system with the check module to distinguish according to the demand from a client system is offered.

[0019] In addition, a check module can be constituted so that it may have the function according to claim 8 which downloads an update module applicable to a client system from a server system like, and the function which applies this update module to a client system.
[0020] According to this updating system, it is possible to realize easily the updating approach of the software which does so the effectiveness which was excellent in the ****.
[0021] Moreover, as for a check module, it is desirable to have the function according to claim 5 to judge whether the configuration file of software can be investigated and this software can be updated like, and the function which notifies the user of a client system of that when it is judged that an update process of software cannot be performed.
[0022] According to this updating approach, a check module investigates the configuration file of software, and even if there is nonconformity -- the indispensable file is lost by software -- it is restorable with the worst user's own hand. For this reason, the software in a client system can be maintained at an always healthy condition.
[0023] Moreover, as for a check module, it is desirable to have the function according to claim 6 to judge the existence of the computer virus of software like.
[0024] According to this updating approach, it is automatically removable even if computer virus exists in the software of a client system. For this reason, the software in a client system can be maintained at an always healthy condition.
[0025] Moreover, as for a check module, it is desirable to unify the update information according to claim 7 collected from the software vendor system like, to create comprehensive update information, and to have the function which distinguishes an update module applicable to a client system based on this comprehensive update information.
[0026] According to this updating system, decision of an update module required for an update process of the software installed in the client system and the configuration file of software can be investigated, and it can judge easily whether this software can be updated.
[0027]
[Embodiment of the Invention] The updating approach of the software concerning this invention and the gestalt of suitable operation of an updating system are explained to a detail, referring to an accompanying drawing below. In addition, in this description and a drawing, duplication explanation is omitted by ****** which attaches the same sign about the component which has the same functional configuration substantially.
[0028] First, 1 operation gestalt of the updating system of the software concerning this invention is explained, referring to drawing 1 . The updating system 10 of software is a system constituted including the client system 100, the server system 106, and software vendor system 116,116', as shown in drawing 1 . These client system 100, the server system 106, and software vendor system 116,116' are mutually connected in that data can be exchanged through networks, such as the Internet.
[0029] (Client system 100) The client system 100 has a store 102 and has 1 or two or more software 104 which are set as the object of update (updating) in a store 102. With the gestalt of this operation, in the client system 100, the program equivalent to the updater component of software is not installed, but it differs from the system conventional at this point. In addition, in drawing 1 , although only one client system 100 is shown, many client systems exist in the updating system 10 actually.
[0030] (Server system 106) The server system 106 has storage 108 and a temporary memory 112. In the store 108, the comprehensive information list 110 of the software for updating the software of the client system 100 is stored. About this comprehensive information list 110, it mentions later further. Moreover, into a temporary memory 112, the cache of the update module 114 of the software of the client system 100 is carried out suitably. The update module 114 of software is offered from software vendor system 116,116'. In addition, two or more such server systems may exist in the updating system 10.
[0031] (Software vendor system 116) software vendor system 116,116' -- respectively -- storage 118,118' -- having -- storage 118,118' -- it has information list 120,120' of software, and update module 122,122' of software inside. About this information list 120,120', it mentions later further. In addition, in drawing 1 , although only software vendor system 116,116' is shown,

many client systems exist in the updating system 10 actually.
[0032] above -- constituting -- having -- software -- updating -- a system -- ten -- depending -- if -- a client -- a system -- 100 -- one -- a ** -- a server -- a system -- 106 -- accessing -- things -- a server -- a system -- 106 -- minding -- all -- software -- a vendor -- a system -- 116,116 -- ' -- providing -- update -- a module -- 122,122 -- ' -- offer -- it can receive .
[0033] subsequently, the comprehensive information list 110 which was stored in the store 108 of the server system 106 in explaining an update process of actual software and the store 118,118 of software vendor system 116,116' -- relation with 'the information list 120,120 stored inside' is explained, referring to drawing 2 .
[0034] The server system 106 creates the comprehensive information list 110 from information list 120,120' of the software which software vendor system 116,116' puts up. The comprehensive information list 110 of software summarizes the information about each software by predetermined format.
[0035] The comprehensive information list 110 has indicated six items of the prerequisite 208 for updating the software vendor name 200 which offers software, the software name 202, the version 204 of software, the configuration file 206 of software, and software, and the update information 210 of software about each software, respectively, as shown in drawing 2 . The page concerned of a website is serially checked for every software vendor, required information is extracted automatically and the comprehensive information list 110 can create it.
[0036] And the server system 106 has a check module offer means to provide the client system 100 with the check module for updating software 104, based on the above-mentioned comprehensive information list 110. This check module is the program which can be performed on the client system 100. Below, processing actuation of a check module is explained, referring to drawing 3 .
[0037] If the client system 100 accesses to the server system 106, a check module will be downloaded promptly and will operate only through the server system 106 as follows.
[0038] (Step S300) To all the files stored in the storage 102 of the client system 100, the successive approximation of each item of the configuration file 206 in the comprehensive information list 110 of software is carried out, and the detail of the software 104 installed in the client system 100 is detected. For example, in the comprehensive information list 110 of drawing 2 , if files A1, B1, and C1 exist in a store 102, it turns out that version vl.0 of software 1 is installed, and if files A1, B1, and D1 exist, it turns out that version vl.1 of software 1 is installed.
[0039] (Step S302) About each of the detected software 104, the detail about the propriety of updating is distinguished with reference to the prerequisite 208 and update information 210 in the comprehensive information list 110 of software. For example, in the comprehensive information list 110 of drawing 2 , it turns out that can update version vl.0 of software 2 to a version v2.0 by applying the update module 1 if OS is vl.0, and it can update to a version v3.0, respectively by applying the update modules 1 and 2 if OS is more than v2.0.
[0040] On the other hand, when the software which can be updated does not exist, all the files relevant to the check module downloaded from the server system 106 are deleted, and processing is ended (step S308).
[0041] (Step S304) About each of the software 104 which can be updated, an applicable update module is downloaded through the server system 106 with reference to the software vendor name 200 in the comprehensive information list 110 of software. To the download demand of the client system 100, the server system 106 will transmit it, if the cache of the update module concerned is carried out into the temporary memory 112. When a cache is not carried out, the update module concerned is downloaded as a deputy from each software vendor system 116,116', and after carrying out a cache into a temporary memory 112, it transmits to the client system 100. That is, the server system 106 has a function as a kind of proxy server which has a cache function.
[0042] (Step S306) The update module downloaded from the server system 106 is serially applied to the software 104 installed in the client system 100. Usually, since an update module is the program which can be performed on the client system 100, a check module should just

perform these update module as a child process.
[0043] (Step S308) All the files relevant to the check module and update module which were downloaded from the server system 106 are deleted, and processing is ended.
[0044] Renewal of the software 104 installed in the storage 102 of the client system 100 as mentioned above is realized.
[0045] As explained above, only when it is not necessary to install the program equivalent to an updater component in the client system 100 in advance and the need for renewal of software arises according to the gestalt of this operation, he is trying to download a check module from the server system 106. And after an update process of software is completed, he is trying to delete all the files relevant to a check module and an update module. For this reason, storage capacity of the client system 100 is not consumed vainly.
[0046] Moreover, since all software installed in the client system 100 by accessing one server system 106 can be updated, while mitigating consumption of a network resource, the effort of renewal of software is mitigable.
[0047] (Gestalt of the 2nd operation) The gestalt of this operation applies the gestalt of the 1st operation of a ****, and adds a new function to a check module. Processing of the check module concerning the gestalt of this operation is explained referring to drawing 4 .
[0048] First, about step S300 which detects the software installed in the client system 100, it is the same as that of the case of the gestalt of implementation of the above 1st.
[0049] (Step S400) In advance of those update processes, the quality of consistency is distinguished to the software 104 installed in the client system 100. Each software means whether it has agreed in the configuration file 206 and prerequisite 208 in the comprehensive information list 110, and the quality of consistency can perform the distinction here, in case the detail of software 104 is detected like step S300 in the gestalt of the 1st operation.
[0050] For example, files A1 and B1 exist in the storage 102 of the client system 100, and suppose that files C1 and D1 were not found. In this case, according to the comprehensive information list 110 of drawing 2 , the newest version when files A1 and B1 exist is the version v1.1 which considers files A1, B1, and D1 as a configuration file. Usually, it is updated by replacing with the configuration file C1 of a version v1.0 (configuration files A1, B1, and C1) to a version v1.1 (configuration files A1, B1, and D1), and adding a file D1 to it. For this reason, when files A1 and B1 exist in storage 102 and files C1 and D1 are not found, it progresses to step S302 as that (namely, thing in which the file C1 exists) in which version vl.0 of software 1 is installed.
[0051] Suppose that files A1 and C1 existed in storage 102, and a file B1 was not found on the other hand. In this case, according to the comprehensive information list 110 of drawing 2 , updating is impossible even if it carries out with what kind of update module. for this reason, restoration of software is impossible -- it flies to the phase (step S308) of deleting all the files that notified the user of the client system 100 of a certain purport (step S402), and downloaded it after that.
[0052] Thus, as a result of distinguishing the quality of consistency, with an update module, if restoration is possible, an update process will be continued, otherwise, the user of the client system 100 is notified of that, and a post process is performed.
[0053] Even the phase (step S308) of deleting all the downloaded files is reached from the phase after step S400, i.e., the phase of judging whether the detected software being updated, (step S302), and it is the same as that of the case of the gestalt of implementation of the above 1st.
[0054] According to the gestalt of this operation, as explained above, even if the important file in a client system is lost by a user's failure etc., it is restored by automatic or hand control, and the software installed in the client system 100 can be maintained at a healthy condition.
[0055] (Gestalt of the 3rd operation) The gestalt of this operation applies the gestalt of the 1st operation of a ****, and adds a new function to a check module. Processing of the check module concerning the gestalt of this operation is explained referring to drawing 5 .
[0056] (Step S500) The gestalt of this operation distinguishes the existence of computer virus in advance of those update processes to the software 104 installed in the computer system 100.
[0057] This distinction builds antivirus software into the check module in advance, and can be

performed by operating antivirus software like the update module of step S306 in the gestalt of the 1st operation. It is removed by this antivirus software when computer virus is discovered by the store 102 of the client system 100 as a result of distinction.
[0058] Even the phase (step S308) of deleting all the downloaded files is reached from the phase after step S500, i.e., the phase of detecting the software installed, (step S300), and it is the same as that of the case of the gestalt of implementation of the above 1st.
[0059] In addition, it is also possible to apply the gestalt of this operation to the gestalt of the 2nd operation. That is, it is also possible to make it have the both sides of the function which investigates the configuration file of software to a check module, and distinguishes the quality of the consistency of software to it, and the function to judge the existence of computer virus.
[0060] According to the gestalt of this operation, even if computer virus exists in the client system 100, it is removed automatically, and the software installed in the client system 100 can be maintained at a healthy condition.
[0061] As mentioned above, although the suitable operation gestalt of the updating approach of the software concerning this invention and an updating system was explained referring to an accompanying drawing, this invention is not limited to this example. If it is this contractor, it will be clear that it can hit on an idea for various kinds of examples of modification or examples of correction in the criteria of the technical thought indicated by the claim, and it will be understood as what naturally belongs to the technical range of this invention also about them.
[0062] For example, this invention is applicable not only to the case of renewal of a client system with an Internet connectivity function but the renewal of the client machine of the client/server system in a Local Area Network.
[0063]
[Effect of the Invention] What is necessary is to download a check module from a server system, only when it is not necessary to install the program equivalent to an updater component in a client system in advance and the need for renewal of software arises according to this invention, as explained above. And after an update process of software is completed, all the files relevant to a check module and an update module can be deleted. For this reason, storage capacity of a client system is not consumed vainly.
[0064] Moreover, since all software installed in the client system by accessing one server system can be updated, while mitigating consumption of a network resource, the effort of renewal of software is mitigable.
[0065] Moreover, a check module investigates the configuration file of software, and since it is restorable with the worst user's own hand even if there is nonconformity -- the indispensable file is lost by software -- the software in a client system can be maintained at an always healthy condition.
[0066] Moreover, since it is automatically removable even if computer virus exists in the software of a client system, the software in a client system can be maintained at an always healthy condition.

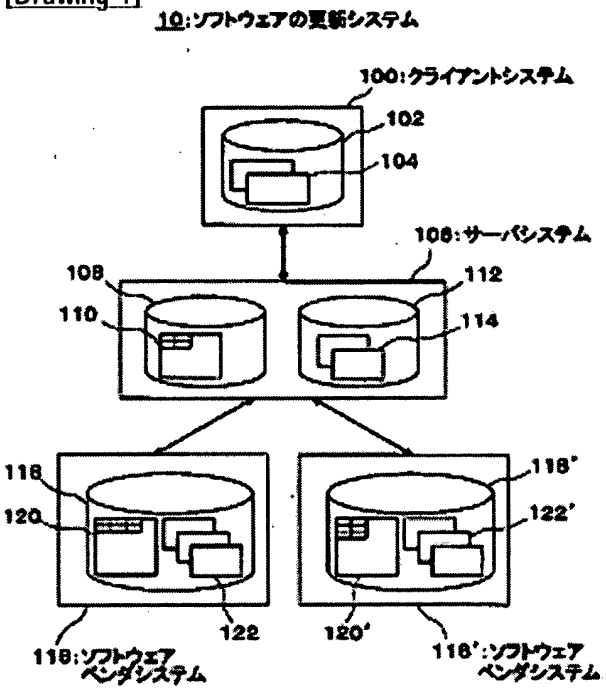
[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

[Drawing 1]

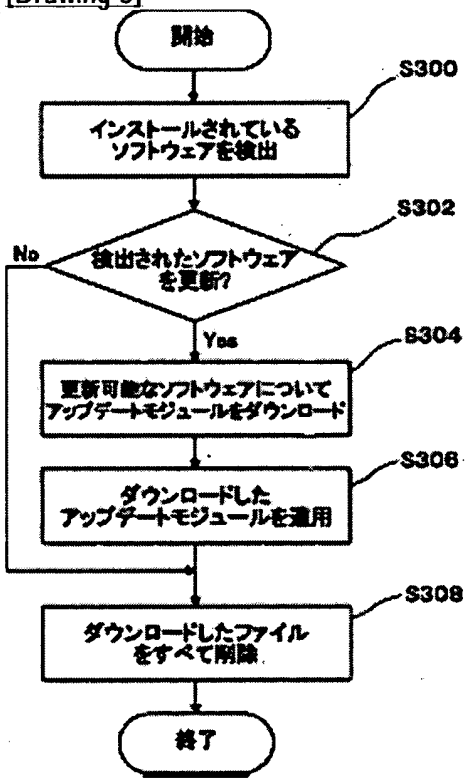


[Drawing 2]

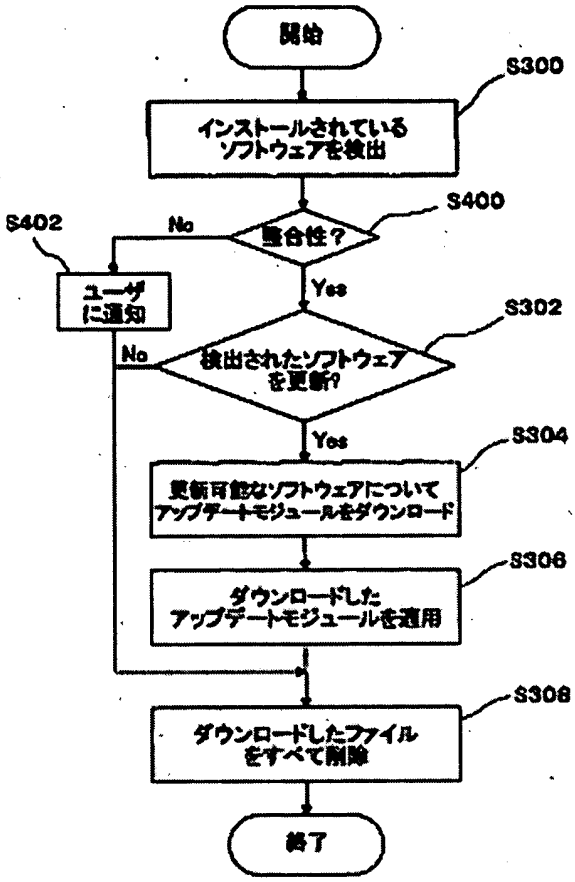
110:ソフトウェアの総合情報リスト

200	202	204	206	208	210
ソフトウェアベンダ名	ソフトウェア名	v1.0	構成ファイル	拡張条件	更新情報
ソフトウェアベンダ1	ソフトウェア1	v1.0	ファイルA1 ファイルB1 ファイルC1	OS v1.0以上	-
		v1.1	ファイルA1 ファイルB1 ファイルD1	OS v1.0以上	v1.0に対しアップデートモジュール1を適用
		v2.0	ファイルA2 ファイルB2 ファイルC2	OS v1.0以上 ソフトウェア2 v2.0	v1.xから更新不可
ソフトウェアベンダ2	ソフトウェア2	v1.0	ファイルX1 ファイルY1 ファイルZ1	OS v1.0以上	-
		v2.0	ファイルX2 ファイルY2 ファイルZ2	OS v1.0以上	v1.0に対しアップデートモジュール4を適用
		v3.0	ファイルX3 ファイルY3 ファイルZ3	OS v2.0以上	v2.0に対しアップデートモジュール5を適用

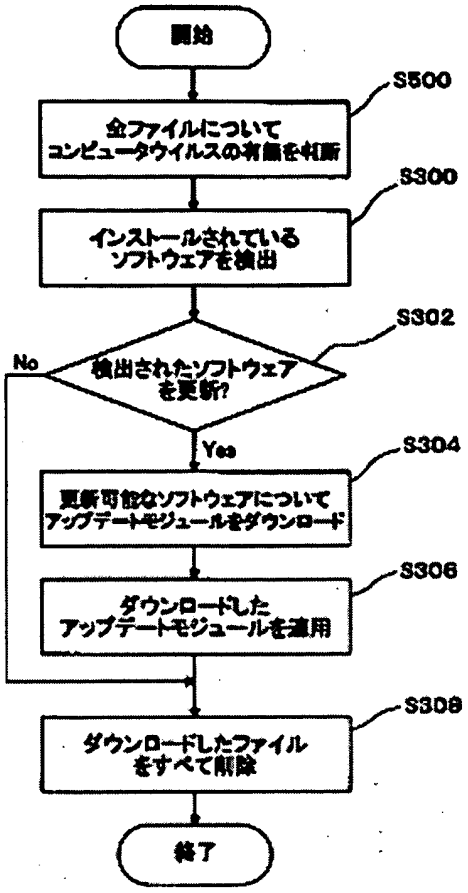
[Drawing 3]



[Drawing 4]


開始
S300
インストールされている
ソフトウェアを検出
S400
整合性?
No
S402
ユーザ
に通知
Yes
S302
検出されたソフトウェア
を更新?
No
Yes
S304
更新可能なソフトウェアについて
アップデートモジュールをダウンロード
S306
ダウンロードした
アップデートモジュールを適用
S308
ダウンロードしたファイル
をすべて削除
終了

[Drawing 5]



[Translation done.]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－7129
（P2002－7129A）
(43)公開日　平成14年１月11日(2002.1.11)

(51)Int.Cl.'	識別記号	ＦＩ		ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０６Ｆ　9/445		Ｇ０６Ｆ　9/06	６１０Ｑ	５Ｂ０７６
11/00			６６０Ｎ	

審査請求　未請求　請求項の数８　ＯＬ　（全９頁）

(21)出願番号　特願2000－184282(P2000－184282)

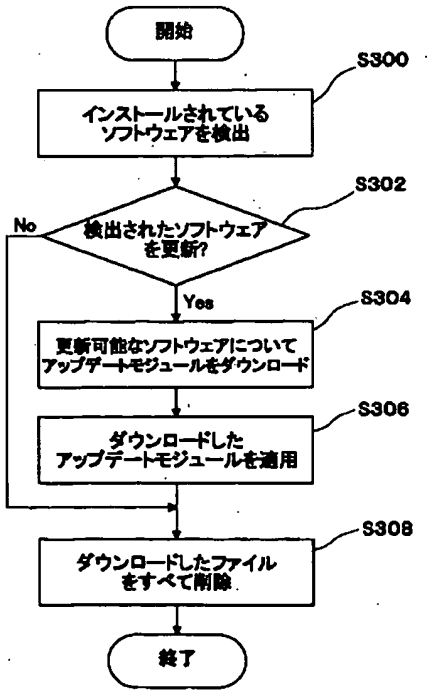
(22)出願日　平成12年６月20日(2000.6.20)

(71)出願人　000000295
沖電気工業株式会社
東京都港区虎ノ門１丁目７番12号
(72)発明者　堀川　慎一
東京都港区虎ノ門１丁目７番12号　沖電気工業株式会社内
(74)代理人　100095957
弁理士　亀谷　美明　（外３名）

Ｆターム(参考)　5B076 AC03 BB06 EA07

(54)【発明の名称】　ソフトウェアの更新方法及び更新システム

(57)【要約】
【課題】　記憶容量を浪費することなくソフトウェアの更新が可能であり，１つのウェブサイトのみへアクセスすることにより，ソフトウェアのアップデートが可能な，ソフトウェアの更新方法及び更新システムを提供する。
【解決手段】　クライアントシステム１００が，サーバシステム１０６にアクセスし，ソフトウェア１０４の更新処理に必要なチェックモジュールをダウンロードする段階と，チェックモジュールにより，クライアントシステムにインストールされたソフトウェアの更新処理に必要なアップデートモジュールを判断する段階（Ｓ３０２）と，クライアントシステムが，サーバシステムにアクセスし，各ソフトウェアの更新処理に必要なアップデートモジュールをダウンロードする段階（Ｓ３０４）と，チェックモジュールとアップデートモジュールをクライアントシステム内より削除する段階（Ｓ３０８）を含む。



1

【特許請求の範囲】

【請求項１】　ネットワークを介してサーバシステムからクライアントシステムへアップデートモジュールをダウンロードし，前記アップデートモジュールを前記クライアントシステムに適用することにより，前記クライアントシステムにインストールされたソフトウェアの更新を行う方法であって，前記クライアントシステムが，前記サーバシステムにアクセスし，ソフトウェアの更新処理に必要なアップデートモジュールを判断するためのチェックモジュールをダウンロードする第１段階と，前記チェックモジュールにより，前記クライアントシステムにインストールされたソフトウェアの更新処理に必要なアップデートモジュールを判断する第２段階と，前記クライアントシステムが，前記サーバシステムにアクセスし，各ソフトウェアの更新処理に必要なアップデートモジュールをダウンロードする第３段階と，前記チェックモジュールと前記アップデートモジュールを前記クライアントシステム内より削除する第４段階と，を含むことを特徴とする，ソフトウェアの更新方法。

【請求項２】　前記第２段階は，前記チェックモジュールが，ソフトウェアの構成ファイルを調べ，該ソフトウェアの更新処理を行うことができるか否かを判断する段階と，前記チェックモジュールが，ソフトウェアの更新処理を行うことができないと判断した場合に，その旨を前記クライアントシステムのユーザに通知する段階と，を含むことを特徴とする，請求項１に記載のソフトウェアの更新方法。

【請求項３】　前記第１段階の後に，前記チェックモジュールにより，各ソフトウェアのコンピュータウイルスの有無を判断する段階を含むことを特徴とする，請求項１または２に記載のソフトウェアの更新方法。

【請求項４】　ネットワークを介してサーバシステムからクライアントシステムへアップデートモジュールをダウンロードし，前記アップデートモジュールを前記クライアントシステムに適用することにより，前記クライアントシステムにインストールされたソフトウェアの更新を行うシステムであって，前記サーバシステムは，前記ソフトウェアの更新に必要な更新情報を，前記アップデートモジュールを提供するソフトウェアベンダシステムから収集し蓄積する第１の記憶手段と，前記クライアントシステムに適用可能なアップデートモジュールを前記ソフトウェアベンダシステムから収集し蓄積する第２の記憶手段と，前記クライアントシステムに適用可能なアップデートモジュールを判別するチェックモジュールを，前記クライアントシステムからの要求に応じて前記クライアントシステムに提供するチェックモジュール提供手段と，を含むことを特徴とする，ソフトウェアの更新システム。

【請求項５】　前記チェックモジュールは，ソフトウェアの構成ファイルを調べ，該ソフトウェアの更新処理を行うことができるか否かを判断する機能と，ソフトウェアの更新処理を行うことができないと判断した場合に，その旨を前記クライアントシステムのユーザに通知する機能とを有することを特徴とする，請求項４に記載のソフトウェアの更新システム。

2

【請求項６】　前記チェックモジュールは，ソフトウェアのコンピュータウイルスの有無を判断する機能を有することを特徴とする，請求項４または５に記載のソフトウェアの更新システム。

【請求項７】　前記チェックモジュールは，前記ソフトウェアベンダシステムから収集した更新情報を統合して，総合更新情報を作成し，該総合更新情報に基づいて，前記クライアントシステムに適用可能なアップデートモジュールを判別する機能を有することを特徴とする，請求項４，５または６のいずれかに記載のソフトウェアの更新システム。

【請求項８】　前記チェックモジュールは，前記クライアントシステムに適用可能なアップデートモジュールを前記サーバシステムからダウンロードする機能と，該アップデートモジュールを前記クライアントシステムに適用する機能とを有することを特徴とする，請求項４，５，６または７のいずれかに記載のソフトウェアの更新システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は，コンピュータシステムにインストールされたソフトウェアの更新（アップデート）を行う更新方法及び更新システムに関する。

【０００２】

【従来の技術】近年，コンピュータシステムの高度化・複雑化に伴い，ソフトウェアの修正や機能強化等を行うアップデートモジュールが頻繁に提供されるようになった。アップデートモジュールの配布手段としてはインターネット上のウェブサイトが広く利用されているが，それらは一般にソフトウェアベンダ毎に存在し，アップデートモジュールに関する情報をソフトウェア製品毎に掲載していることが多い。

【０００３】一方，ソフトウェア及びそのアップデートモジュールの提供を受けるコンピュータシステム（以下，「クライアントシステム」と称する。）には複数のソフトウェアがインストールされており，異なるソフトウェアベンダのソフトウェア同士が依存関係にある場合も少なくない。従って，クライアントシステムにおいては，適用すべきアップデートモジュールの取捨選択に多大な労力が必要である。

【０００４】このような問題を解決する従来技術としては，文献「シーマス・ドノヒュー：" コンピュータ・ネットワークを通してソフトウェア更新を配布する装置" ，特開平１１－２７２４５４」に開示されるものがある。この文献で示された方法では，クライアントシス

テムに，更新対象となるソフトウェアとともにアップデータ・コンポーネントがインストールされており，かかるアップデータ・コンポーネントにより，以下のようにしてソフトウェアの更新が行われる。

【０００５】（１）クライアントシステムにインストールされた各ソフトウェアに対し，ネットワーク上に用意されたサーチ・エンジンを介してソフトウェアベンダシステムを検索する。

（２）それぞれのウェブサイトから更新リストをダウンロードし，これらの更新リストを基に各ソフトウェアの更新の可否を判別する。

（３）更新可能なソフトウェアについて，そのソフトウェアベンダシステムからアップデートモジュールをダウンロードし，クライアントシステムへ適用する。

【０００６】上記文献では，例としてＵＮＩＸ（登録商標）系ＯＳ（Ｏｐｅｒａｔｉｎｇ　Ｓｙｓｔｅｍ）の場合が取り上げられているが，Ｗｉｎｄｏｗｓ（登録商標）系ＯＳにおける同種の技術には，マイクロソフト社の「Ｗｉｎｄｏｗｓ　Ｕｐｄａｔｅ」やシマンテック社の「Ｌｉｖｅ　Ｕｐｄａｔｅ」が存在する。これらはともに各社が提供するソフトウェア製品のみを対象としたものであるが，サーチエンジンを用いない点を除いて上記文献と同様な処理によりソフトウェアの更新が行われる。

【０００７】

【発明が解決しようとする課題】ところで，上記従来技術においては，異なるソフトウェアベンダが提供する複数のソフトウェアの更新を行う場合，各ソフトウェアごとに対応するソフトウェアベンダシステムを判断してアクセスする必要がある。これは，限られたネットワーク資源の有効活用の観点からすると，望ましいものではない。また，クライアントシステムに多数のソフトウェアがインストールされている場合には，これら多数のソフトウェアの更新の必要性を常に把握しておかなければならず，不便きわまりない。

【０００８】従来技術ではまた，ソフトウェアの更新に関わるアップデータ・コンポーネントがいずれも事前にクライアントシステムへインストールされる点が共通する。このアップデータ・コンポーネントは，クライアントシステム全体の進歩に伴いそれ自身を更新しなければならないことがあり，記憶装置の容量を無駄に消費するだけの存在になりかねない。

【０００９】本発明は，従来のソフトウェアの更新システムが有する上記問題点に鑑みてなされたものであり，本発明の目的は，クライアントシステムの記憶容量を浪費することなくソフトウェアの更新が可能な，新規かつ改良されたソフトウェアの更新システムを提供することである。

【００１０】また，本発明の他の目的は，１つのウェブサイトのみへアクセスすることにより，クライアントシステム内のソフトウェアのアップデートが可能な，新規かつ改良されたソフトウェアの更新システムを提供することである。

【００１１】

【課題を解決するための手段】上記課題を解決するため，本発明の第１の観点によれば，請求項１に記載のように，ネットワークを介してサーバシステムからクライアントシステムへアップデートモジュールをダウンロードし，アップデートモジュールをクライアントシステムに適用することにより，クライアントシステムにインストールされたソフトウェアの更新を行う方法であって，クライアントシステムが，サーバシステムにアクセスし，ソフトウェアの更新処理に必要なアップデートモジュールを判断するためのチェックモジュールをダウンロードする第１段階と，チェックモジュールにより，クライアントシステムにインストールされたソフトウェアの更新処理に必要なアップデートモジュールを判断する第２段階と，クライアントシステムが，サーバシステムにアクセスし，各ソフトウェアの更新処理に必要なアップデートモジュールをダウンロードする第３段階と，チェックモジュールとアップデートモジュールをクライアントシステム内より削除する第４段階とを含むことを特徴とするソフトウェアの更新方法が提供される。

【００１２】かかる更新方法によれば，アップデータ・コンポーネントに相当するプログラムをクライアントシステムに事前にインストールしておく必要がなく，ソフトウェアの更新の必要が生じた場合にのみ，サーバシステムよりチェックモジュールをダウンロードすればよい。そして，ソフトウェアの更新処理が終了した後には，チェックモジュール及びアップデートモジュールに関連するすべてのファイルを削除することができる。このため，クライアントシステムの記憶容量を無駄に消費することがない。

【００１３】また，１つのサーバシステムにアクセスすることにより，クライアントシステムにインストールされたすべてのソフトウェアの更新を行うことができるので，ネットワーク資源の消費を軽減するとともにソフトウェアの更新の労力を軽減することができる。

【００１４】また，第２段階は，請求項２に記載のように，チェックモジュールが，ソフトウェアの構成ファイルを調べ，該ソフトウェアの更新処理を行うことができるか否かを判断する段階と，チェックモジュールが，ソフトウェアの更新処理を行うことができないと判断した場合に，その旨をクライアントシステムのユーザに通知する段階とを含むことが好ましい。

【００１５】かかる更新方法によれば，チェックモジュールがソフトウェアの構成ファイルを調べ，ソフトウェアに必須のファイルが失われている等の不具合があっても，最悪ユーザ自身の手により修復することができる。このため，クライアントシステム内のソフトウェアを常

に健全な状態に保つことができる。

【００１６】また，請求項３に記載のように，第１段階の後に，チェックモジュールにより，各ソフトウェアのコンピュータウイルスの有無を判断する段階を含むことが好ましい。

【００１７】かかる更新方法によれば，クライアントシステムのソフトウェアにコンピュータウイルスが存在しても，自動的に除去することができる。このため，クライアントシステム内のソフトウェアを常に健全な状態に保つことができる。

【００１８】また，上記課題を解決するため，本発明の第２の観点によれば，請求項４に記載のように，ネットワークを介してサーバシステムからクライアントシステムへアップデートモジュールをダウンロードし，アップデートモジュールをクライアントシステムに適用することにより，クライアントシステムにインストールされたソフトウェアの更新を行うシステムであって，サーバシステムは，ソフトウェアの更新に必要な更新情報を，アップデートモジュールを提供するソフトウェアベンダシステムから収集し蓄積する第１の記憶手段と，クライアントシステムに適用可能なアップデートモジュールをソフトウェアベンダシステムから収集し蓄積する第２の記憶手段と，クライアントシステムに適用可能なアップデートモジュールを判別するチェックモジュールを，クライアントシステムからの要求に応じてクライアントシステムに提供するチェックモジュール提供手段とを含むことを特徴とするソフトウェアの更新システムが提供される。

【００１９】なお，チェックモジュールは，請求項８に記載のように，クライアントシステムに適用可能なアップデートモジュールをサーバシステムからダウンロードする機能と，該アップデートモジュールをクライアントシステムに適用する機能とを有するように構成することができる。

【００２０】かかる更新システムによれば，上述の優れた効果を奏するソフトウェアの更新方法を容易に実現することが可能である。

【００２１】また，チェックモジュールは，請求項５に記載のように，ソフトウェアの構成ファイルを調べ，該ソフトウェアの更新処理を行うことができるか否かを判断する機能と，ソフトウェアの更新処理を行うことができないと判断した場合に，その旨をクライアントシステムのユーザに通知する機能とを有することが好ましい。

【００２２】かかる更新方法によれば，チェックモジュールがソフトウェアの構成ファイルを調べ，ソフトウェアに必須のファイルが失われている等の不具合があっても，最悪ユーザ自身の手により修復することができる。このため，クライアントシステム内のソフトウェアを常に健全な状態に保つことができる。

【００２３】また，チェックモジュールは，請求項６に記載のように，ソフトウェアのコンピュータウイルスの有無を判断する機能を有することが好ましい。

【００２４】かかる更新方法によれば，クライアントシステムのソフトウェアにコンピュータウイルスが存在しても，自動的に除去することができる。このため，クライアントシステム内のソフトウェアを常に健全な状態に保つことができる。

【００２５】また，チェックモジュールは，請求項７に記載のように，ソフトウェアベンダシステムから収集した更新情報を統合して，総合更新情報を作成し，該総合更新情報に基づいて，クライアントシステムに適用可能なアップデートモジュールを判別する機能を有することが好ましい。

【００２６】かかる更新システムによれば，クライアントシステムにインストールされたソフトウェアの更新処理に必要なアップデートモジュールの判断や，ソフトウェアの構成ファイルを調べ，該ソフトウェアの更新処理を行うことができるか否かの判断を容易に行うことができる。

【００２７】

【発明の実施の形態】以下に添付図面を参照しながら，本発明にかかるソフトウェアの更新方法及び更新システムの好適な実施の形態について詳細に説明する。なお，本明細書及び図面において，実質的に同一の機能構成を有する構成要素については，同一の符号を付することにより重複説明を省略する。

【００２８】まず，本発明にかかるソフトウェアの更新システムの一実施形態について，図１を参照しながら説明する。ソフトウェアの更新システム１０は，図１に示したように，クライアントシステム１００と，サーバシステム１０６と，ソフトウェアベンダシステム１１６，１１６’とを含んで構成されるシステムである。これら，クライアントシステム１００，サーバシステム１０６，ソフトウェアベンダシステム１１６，１１６’は，インターネット等のネットワークを介してデータをやりとりできるという点で，相互に接続されている。

【００２９】（クライアントシステム１００）クライアントシステム１００は，記憶装置１０２を有し，記憶装置１０２内に，アップデート（更新）の対象となる１または２以上のソフトウェア１０４を持つ。本実施の形態では，クライアントシステム１００内には，ソフトウェアのアップデータ・コンポーネントに相当するプログラムがインストールされておらず，この点で従来のシステムと異なる。なお，図１では，１つのクライアントシステム１００のみを示しているが，実際には，多数のクライアントシステムが更新システム１０内に存在する。

【００３０】（サーバシステム１０６）サーバシステム１０６は，記憶装置１０８と一時記憶装置１１２とを有している。記憶装置１０８内には，クライアントシステム１００のソフトウェアをアップデートするためのソフ

トウェアの総合情報リスト１１０が格納されている。この総合情報リスト１１０については，さらに後述する。また，一時記憶装置１１２内には，クライアントシステム１００のソフトウェアのアップデートモジュール１１４が適宜キャッシュされる。ソフトウェアのアップデートモジュール１１４は，ソフトウェアベンダシステム１１６，１１６’より提供される。なお，このようなサーバシステムは，更新システム１０内に複数存在してもよい。

【００３１】（ソフトウェアベンダシステム１１６）ソフトウェアベンダシステム１１６，１１６’は，それぞれ記憶装置１１８，１１８’を有し，記憶装置１１８，１１８’内に，ソフトウェアの情報リスト１２０，１２０’，及び，ソフトウェアのアップデートモジュール１２２，１２２’を持つ。この情報リスト１２０，１２０’については，さらに後述する。なお，図１では，ソフトウェアベンダシステム１１６，１１６’のみを示しているが，実際には，多数のクライアントシステムが更新システム１０内に存在する。

【００３２】上述のように構成されるソフトウェアの更新システム１０によれば，クライアントシステム１００は，１つのサーバシステム１０６へアクセスすることにより，サーバシステム１０６を介して，すべてのソフトウェアベンダシステム１１６，１１６’が提供するアップデートモジュール１２２，１２２’の提供を受けることができる。

【００３３】次いで，実際のソフトウェアの更新処理を説明するにあたり，サーバシステム１０６の記憶装置１０８内に格納された総合情報リスト１１０と，ソフトウェアベンダシステム１１６，１１６’の記憶装置１１８，１１８’内に格納された情報リスト１２０，１２０’との関係について，図２を参照しながら説明する。

【００３４】サーバシステム１０６は，ソフトウェアベンダシステム１１６，１１６’が掲示するソフトウェアの情報リスト１２０，１２０’から総合情報リスト１１０を作成する。ソフトウェアの総合情報リスト１１０は，各ソフトウェアに関する情報を所定のフォーマットによりまとめたものである。

【００３５】総合情報リスト１１０は，図２に示したように，各ソフトウェアについて，ソフトウェアを提供するソフトウェアベンダ名２００，ソフトウェア名２０２，ソフトウェアのバージョン２０４，ソフトウェアの構成ファイル２０６，ソフトウェアをアップデートするための前提条件２０８，ソフトウェアの更新情報２１０の６項目をそれぞれ記載している。総合情報リスト１１０は，ソフトウェアベンダ毎にウェブサイトの当該ページを逐次確認し，必要な情報を自動的に抽出し作成することが可能である。

【００３６】そして，サーバシステム１０６は，上述の総合情報リスト１１０を基に，ソフトウェア１０４の更新を行うためのチェックモジュールをクライアントシステム１００に提供するチェックモジュール提供手段を有している。このチェックモジュールは，クライアントシステム１００上で実行可能なプログラムである。以下に，図３を参照しながら，チェックモジュールの処理動作について説明する。

【００３７】チェックモジュールは，クライアントシステム１００がサーバシステム１０６へアクセスすると直ちにダウンロードされ，以下のようにサーバシステム１０６のみを介して動作する。

【００３８】（ステップＳ３００）クライアントシステム１００の記憶装置１０２に格納された全ファイルに対し，ソフトウェアの総合情報リスト１１０における構成ファイル２０６の各項を逐次比較して，クライアントシステム１００にインストールされているソフトウェア１０４の詳細を検出する。例えば図２の総合情報リスト１１０の場合，記憶装置１０２にファイルＡ１，Ｂ１，Ｃ１が存在すればソフトウェア１のバージョンｖ１．０がインストールされていることがわかり，ファイルＡ１，Ｂ１，Ｄ１が存在すればソフトウェア１のバージョンｖ１．１がインストールされていることがわかる。

【００３９】（ステップＳ３０２）検出されたソフトウェア１０４のそれぞれについて，ソフトウェアの総合情報リスト１１０における前提条件２０８および更新情報２１０を参照し，更新の可否に関する詳細を判別する。例えば図２の総合情報リスト１１０の場合，ソフトウェア２のバージョンｖ１．０はＯＳがｖ１．０であればアップデートモジュール１を適用することによりバージョンｖ２．０へ更新可能であり，ＯＳがｖ２．０以上であればアップデートモジュール１と２を適用することによりバージョンｖ３．０へそれぞれ更新可能であることがわかる。

【００４０】一方，更新可能なソフトウェアが存在しない場合は，サーバシステム１０６からダウンロードしたチェックモジュールに関連するすべてのファイルを削除し，処理を終了する（ステップＳ３０８）。

【００４１】（ステップＳ３０４）更新可能なソフトウェア１０４のそれぞれについて，ソフトウェアの総合情報リスト１１０におけるソフトウェアベンダ名２００を参照し，適用可能なアップデートモジュールをサーバシステム１０６を介してダウンロードする。サーバシステム１０６は，クライアントシステム１００のダウンロード要求に対し，当該アップデートモジュールが一時記憶装置１１２内にキャッシュされていればそれを転送する。キャッシュされていない場合には，当該アップデートモジュールを各ソフトウェアベンダシステム１１６，１１６’から代理としてダウンロードし，一時記憶装置１１２内にキャッシュした後クライアントシステム１００へ転送する。すなわち，サーバシステム１０６は，キャッシュ機能を有する一種のプロキシサーバとしての機

能を併せ持つ。

【００４２】（ステップＳ３０６）クライアントシステム１００にインストールされたソフトウェア１０４に対し，サーバシステム１０６からダウンロードしたアップデートモジュールを逐次適用する。通常，アップデートモジュールは，クライアントシステム１００上で実行可能なプログラムであるため，チェックモジュールは，それらアップデートモジュールを子プロセスとして実行させれば良い。

【００４３】（ステップＳ３０８）サーバシステム１０６からダウンロードしたチェックモジュールやアップデートモジュールに関連するすべてのファイルを削除し，処理を終了する。

【００４４】以上のようにして，クライアントシステム１００の記憶装置１０２にインストールされたソフトウェア１０４の更新が実現される。

【００４５】以上説明したように，本実施の形態によれば，アップデータ・コンポーネントに相当するプログラムをクライアントシステム１００に事前にインストールしておく必要がなく，ソフトウェアの更新の必要が生じた場合にのみ，サーバシステム１０６よりチェックモジュールをダウンロードするようにしている。そして，ソフトウェアの更新処理が終了した後には，チェックモジュール及びアップデートモジュールに関連するすべてのファイルを削除するようにしている。このため，クライアントシステム１００の記憶容量を無駄に消費することがない。

【００４６】また，１つのサーバシステム１０６にアクセスすることにより，クライアントシステム１００にインストールされたすべてのソフトウェアの更新を行うことができるので，ネットワーク資源の消費を軽減するとともにソフトウェアの更新の労力を軽減することができる。

【００４７】（第２の実施の形態）本実施の形態は，上述の第１の実施の形態を応用したものであり，チェックモジュールに新たな機能を追加したものである。本実施の形態にかかるチェックモジュールの処理を，図４を参照しながら説明する。

【００４８】まず，クライアントシステム１００にインストールされているソフトウェアを検出するステップＳ３００については，上記第１の実施の形態の場合と同様である。

【００４９】（ステップＳ４００）クライアントシステム１００にインストールされたソフトウェア１０４に対し，それらの更新処理に先立って，整合性の良否を判別する。ここで整合性の良否とは，各ソフトウェアが，総合情報リスト１１０における構成ファイル２０６および前提条件２０８に合致しているか否かを意味し，その判別は，第１の実施の形態におけるステップＳ３００と同様にしてソフトウェア１０４の詳細を検出する際に行うことができる。

【００５０】例えば，クライアントシステム１００の記憶装置１０２にファイルＡ１，Ｂ１が存在し，ファイルＣ１，Ｄ１が見当たらなかったとする。この場合，図２の総合情報リスト１１０によれば，ファイルＡ１，Ｂ１が存在しているときの最新のバージョンは，ファイルＡ１，Ｂ１，Ｄ１を構成ファイルとするバージョンｖ１．１である。通常，バージョンｖ１．１（構成ファイルＡ１，Ｂ１，Ｄ１）へは，バージョンｖ１．０（構成ファイルＡ１，Ｂ１，Ｃ１）の構成ファイルＣ１に代えてファイルＤ１が追加されることにより更新される。このため，記憶装置１０２にファイルＡ１，Ｂ１が存在しファイルＣ１，Ｄ１が見当たらない場合には，ソフトウェア１のバージョンｖ１．０がインストールされているもの（すなわち，ファイルＣ１が存在しているもの）としてステップＳ３０２へ進む。

【００５１】一方，記憶装置１０２にファイルＡ１，Ｃ１が存在しファイルＢ１が見当たらなかったとする。この場合は，図２の総合情報リスト１１０によれば，いかなるアップデートモジュールをもってしても更新は不可能である。このため，ソフトウェアの修復が不可能ある旨をクライアントシステム１００のユーザに通知し（ステップＳ４０２），その後，ダウンロードしたファイルをすべて削除する段階（ステップＳ３０８）へ飛ぶ。

【００５２】このように整合性の良否を判別した結果，アップデートモジュールにより修復が可能であれば更新処理を続行し，そうでなければクライアントシステム１００のユーザにその旨を通知して終了処理を行う。

【００５３】ステップＳ４００以降の段階，すなわち，検出されたソフトウェアを更新するか否かを判断する段階（ステップＳ３０２）から，ダウンロードしたファイルをすべて削除する段階（ステップＳ３０８）までについては，上記第１の実施の形態の場合と同様である。

【００５４】以上説明したように，本実施の形態によれば，ユーザの操作ミス等により，クライアントシステム内の重要なファイルが失われていても，自動または手動により修復され，クライアントシステム１００にインストールされたソフトウェアを健全な状態に保つことができる。

【００５５】（第３の実施の形態）本実施の形態は，上述の第１の実施の形態を応用したものであり，チェックモジュールに新たな機能を追加したものである。本実施の形態にかかるチェックモジュールの処理を，図５を参照しながら説明する。

【００５６】（ステップＳ５００）本実施の形態は，コンピュータシステム１００にインストールされたソフトウェア１０４に対し，それらの更新処理に先立ってコンピュータウイルスの有無を判別するものである。

【００５７】この判別は，事前にアンチウイルスソフトウェアをチェックモジュールに組み込んでおき，第１の

実施の形態におけるステップＳ３０６のアップデートモジュールと同様にしてアンチウイルスソフトウェアを動作させることにより行うことができる。判別の結果，クライアントシステム１００の記憶装置１０２にコンピュータウイルスが発見された場合は，このアンチウイルスソフトウェアによって除去される。

【００５８】ステップＳ５００以降の段階，すなわち，インストールされているソフトウェアを検出する段階（ステップＳ３００）から，ダウンロードしたファイルをすべて削除する段階（ステップＳ３０８）までについては，上記第１の実施の形態の場合と同様である。

【００５９】なお，本実施の形態を，第２の実施の形態に適用することも可能である。すなわち，チェックモジュールに，ソフトウェアの構成ファイルを調べ，ソフトウェアの整合性の良否を判別する機能，及び，コンピュータウイルスの有無を判断する機能の双方を有するようにすることも可能である。

【００６０】本実施の形態によれば，クライアントシステム１００内にコンピュータウイルスが存在しても自動的に除去され，クライアントシステム１００にインストールされたソフトウェアを健全な状態に保つことができる。

【００６１】以上，添付図面を参照しながら本発明にかかるソフトウェアの更新方法及び更新システムの好適な実施形態について説明したが，本発明はかかる例に限定されない。当業者であれば，特許請求の範囲に記載された技術的思想の範疇内において各種の変更例または修正例に想到し得ることは明らかであり，それらについても当然に本発明の技術的範囲に属するものと了解される。

【００６２】例えば，本発明は，インターネット接続機能を持つクライアントシステムの更新の場合に限らず，ローカルエリアネットワークにおけるクライアント・サーバシステムのクライアントマシンの更新にも適用することができる。

【００６３】

【発明の効果】以上説明したように，本発明によれば，アップデータ・コンポーネントに相当するプログラムをクライアントシステムに事前にインストールしておく必要がなく，ソフトウェアの更新の必要が生じた場合にのみ，サーバシステムよりチェックモジュールをダウンロードすればよい。そして，ソフトウェアの更新処理が終了した後には，チェックモジュール及びアップデートモジュールに関連するすべてのファイルを削除することができる。このため，クライアントシステムの記憶容量を無駄に消費することがない。

【００６４】また，１つのサーバシステムにアクセスすることにより，クライアントシステムにインストールされたすべてのソフトウェアの更新を行うことができるので，ネットワーク資源の消費を軽減するとともにソフトウェアの更新の労力を軽減することができる。

【００６５】また，チェックモジュールがソフトウェアの構成ファイルを調べ，ソフトウェアに必須のファイルが失われている等の不具合があっても，最悪ユーザ自身の手により修復することができるため，クライアントシステム内のソフトウェアを常に健全な状態に保つことができる。

【００６６】また，クライアントシステムのソフトウェアにコンピュータウイルスが存在しても，自動的に除去することができるため，クライアントシステム内のソフトウェアを常に健全な状態に保つことができる

【図面の簡単な説明】

【図１】ソフトウェアの更新システムの一実施形態を示す説明図である。

【図２】ソフトウェアの総合情報リストを示す説明図である。

【図３】第１の実施の形態にかかるチェックモジュールの処理ブロック図である。

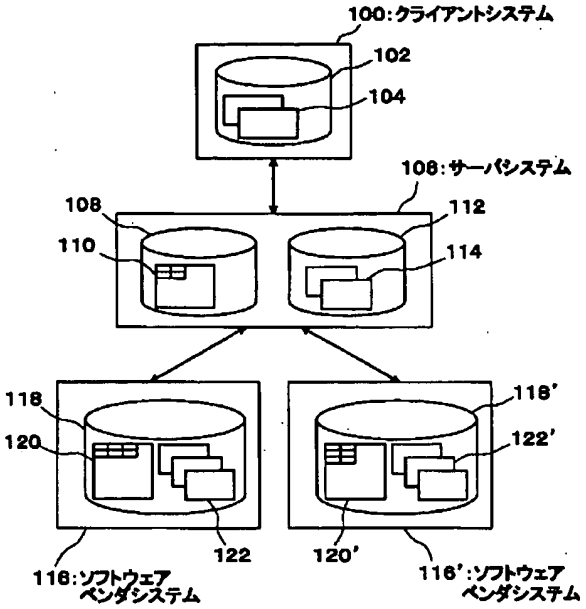
【図４】第２の実施の形態にかかるチェックモジュールの処理ブロック図である。

【図５】第３の実施の形態にかかるチェックモジュールの処理ブロック図である。

【符号の説明】

１０　ソフトウェアの更新システム
１００　クライアントシステム
１０２　記憶装置
１０４　ソフトウェア
１０６　サーバシステム
１０８　記憶装置
１１０　総合情報リスト
１１２　一時記憶装置
１１４　アップデートモジュール
１１６，１１６’　ソフトウェアベンダシステム
１１８，１１８’　記憶装置
１２０，１２０’　情報リスト
１２２，１２２’　アップデートモジュール

【図１】

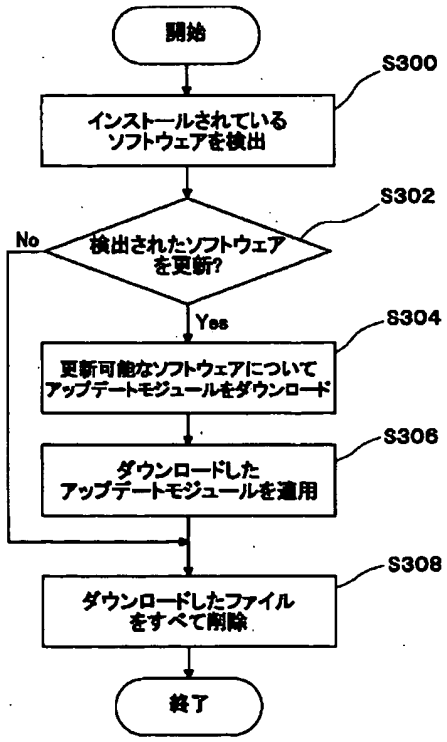


【図２】

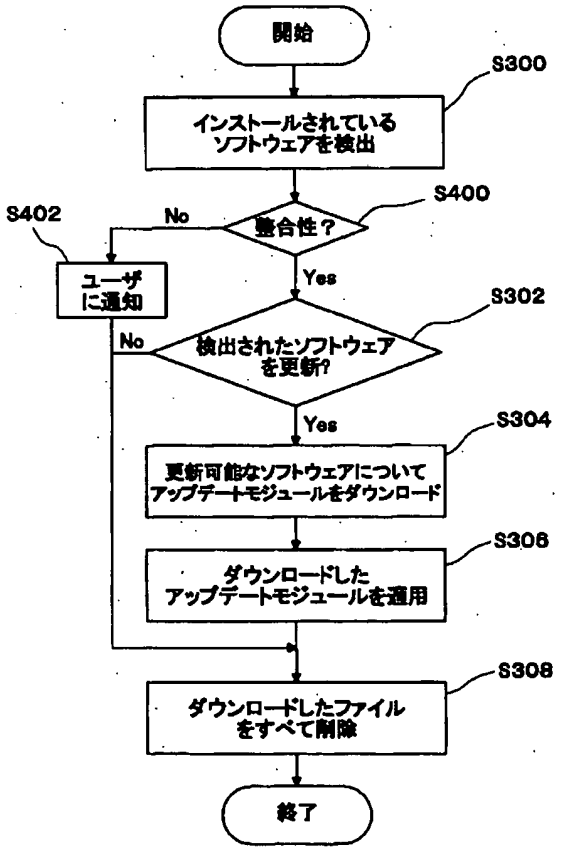
110:ソフトウェアの総合情報リスト

200 ソフトウェアベンダ名	202 ソフトウェア名	204 v1.0	206 構成ファイル	208 前提条件	210 更新情報
ソフトウェアベンダ1	ソフトウェア1	v1.0	ファイルA1 ファイルB1 ファイルC1	OS v1.0以上	－
		v1.1	ファイルA1 ファイルB1 ファイルD1	OS v1.0以上	v1.0に対しアップデートモジュール1を適用
		v2.0	ファイルA2 ファイルB2 ファイルC2	OS v1.0以上 ソフトウェア2 v2.0	v1.xから更新不可
ソフトウェアベンダ2	ソフトウェア2	v1.0	ファイルX1 ファイルY1 ファイルZ1	OS v1.0以上	－
		v2.0	ファイルX2 ファイルY2 ファイルZ2	OS v1.0以上	v1.0に対しアップデートモジュール1を適用
		v3.0	ファイルX3 ファイルY3 ファイルZ3	OS v2.0以上	v2.0に対しアップデートモジュール2を適用

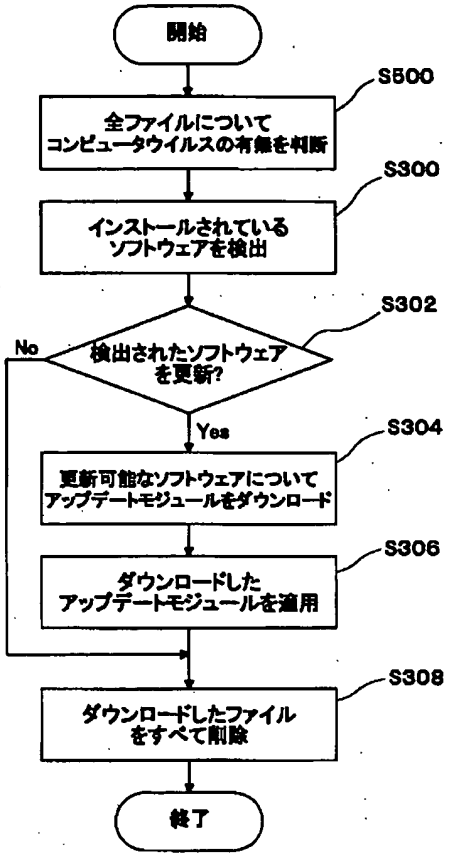
【図３】



【図４】



【図５】


開始
S500
全ファイルについて
コンピュータウイルスの有無を判断
S300
インストールされている
ソフトウェアを検出
S302
No
検出されたソフトウェア
を更新?
Yes
S304
更新可能なソフトウェアについて
アップデートモジュールをダウンロード
S306
ダウンロードした
アップデートモジュールを適用
S308
ダウンロードしたファイル
をすべて削除
終了